

DartBot: Overhand Throwing of Deformable Objects with Tactile Sensing and Reinforcement Learning

Shoaib Aslam*, Krish Kumar*, Pokuang Zhou*, Hongyu Yu, Michael Yu Wang, *Fellow, IEEE*, and Yu She

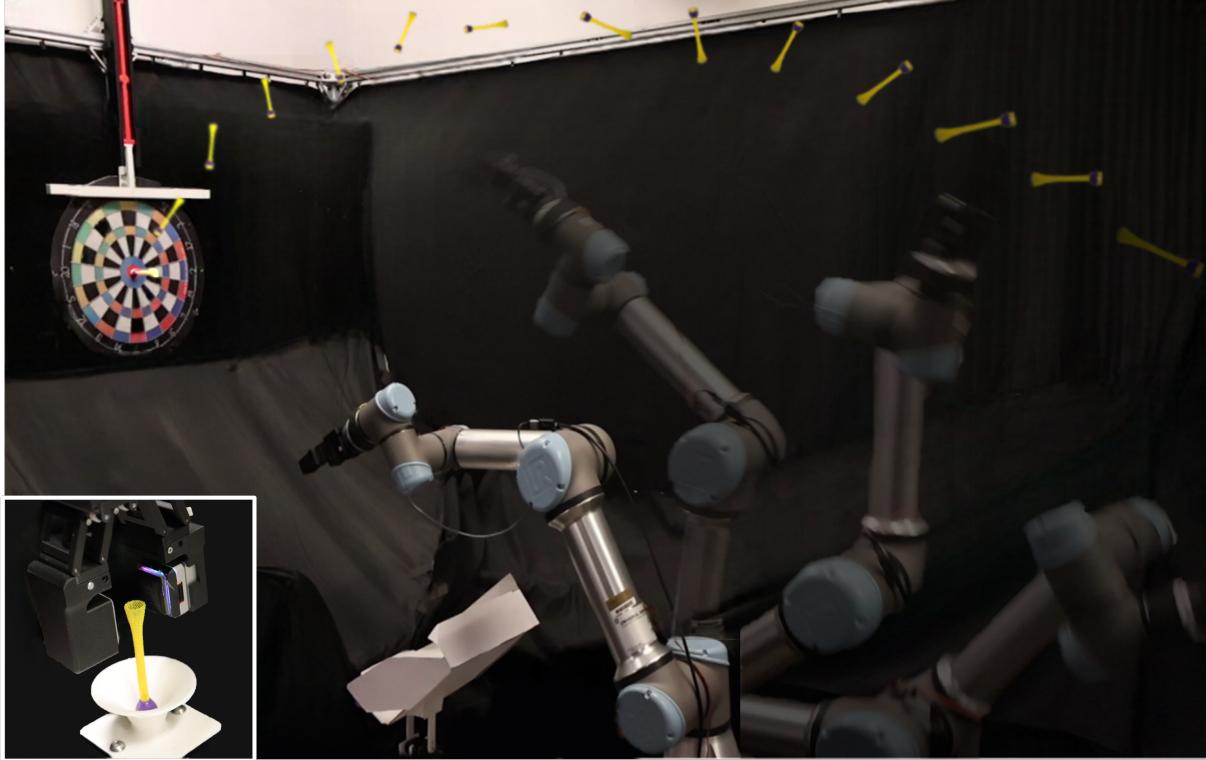


Fig. 1: Throwing object with precision: an example scenario of throwing a nonrigid cylindrical object beyond a robot’s workspace by executing an overhand throw trajectory while object spins in the air (after release) due to its generated moment of inertia. Achieving success in this task, we develop a reinforcement learning-based method with a GelSight Mini tactile sensor, that requires the robot to utilize tactile feedback to perceive the physical properties of the object and deduce the optimal parameters needed to accurately throw the object towards the distant target location. The bottom left image depicts the gripper with the tactile sensor grasping the non-rigid object at its deformable shaft.

Abstract—Object transfer through throwing is a classic dynamic manipulation task that necessitates precise control and perception capabilities. However, developing dynamic models for unstructured environments using analytical methods presents challenges. In this study, we present DartBot, a robot that integrates tactile exploration and reinforcement learning to achieve robust throwing skills for nonrigid objects under the influence of moment of inertia which cause the object to spin in the air. Unlike traditional sim-to-real transfer methods, our approach involves direct training of the agent

on a real hardware robot equipped with a high-resolution tactile sensor, enabling reinforced learning in a realistic and dynamic environment. By leveraging tactile perception, we incorporate pseudo-embeddings of the physical properties of objects into the learning process through tilting actions at two distinct angles. This tactile information enables the agent to infer and adapt its throwing strategy, resulting in improved accuracy when handling various objects and targeting distant locations. Furthermore, we demonstrate that the quality of a grasp significantly impacts the success rate of the throwing task. We evaluate the effectiveness of our method through extensive experiments, demonstrating superior performance and generalization capabilities in real-world throwing scenarios. We achieved a success rate of 95% for unseen objects with a mean error of 3.15 cm from the goal.

* Authors with equal contribution.

Shoaib Aslam, Pokuang Zhou, and Yu She are with the School of Industrial Engineering, Purdue University, West Lafayette, USA (email: saslam@purdue.edu; zhou1458@purdue.edu; shey@purdue.edu). Shoaib Aslam is also with the Department of Mechanical and Aerospace Engineering, The Hong Kong University of Science and Technology, Hong Kong (email: saslamaa@connect.ust.hk). (*Corresponding author: Yu She*)

Krish Kumar is with the Department of Computer Engineering, Purdue University, West Lafayette, USA (email: kumar585@purdue.edu).

Hongyu Yu is with the Department of Mechanical and Aerospace Engineering, The Hong Kong University of Science and Technology, Hong Kong (email: hongyuyu@ust.hk).

Michael Yu Wang is with the School of Engineering, Great Bay University, Dongguan, China (email: mywang@gbu.edu.cn).

Note to Practitioners—The industrial demand for precise and accurate object transfer beyond the robot’s maximum kinematic range is rapidly growing, necessitating advancements in robotic throwing manipulation to efficiently utilize resources. In this context, this paper contributes to the field by presenting a method for the transfer of deformable objects through overhand throwing, addressing the challenges of achieving precise control and perception capabilities by learning the complex physics of

the task using raw data. The approach offers valuable insights and techniques for enhancing throwing manipulation tasks. The integration of high-resolution tactile sensing and reinforcement learning, while considering the influence of moment of inertia, opens up new possibilities for handling deformable objects in throwing tasks and developing dynamic models for such unstructured environments. The learned throwing policy is applied to a variety of previously unseen objects, demonstrating its effectiveness and ability to generalize in real-world throwing scenarios. The presented work holds applicability in various domains such as medical rehabilitation, logistic warehouses for packaging, and handling of urban waste. It shows promise in improving the precision, accuracy and efficiency of the throwing task. In the future, we aim to expand the current overhand throwing framework by incorporating rapid grasping in cluttered environments and enabling throwing to far distances with diverse target locations.

Index Terms—Tactile-based manipulation, force and tactile sensing, perception for grasping and manipulation, reinforcement learning

I. INTRODUCTION

Throwing manipulation, a fundamental aspect of human motor skills [1], plays a critical role in various domains, ranging from sports to industrial automation [2], [3], [4]. The capability to throw empowers a robot to transport objects to a location that surpasses its maximum kinematic limits by harnessing dynamic extrinsic dexterity. However, acquiring the necessary dexterous skills to accurately throw arbitrary objects to specific target locations is a challenging task that depends on various factors. These factors include the physical attributes of the object, such as mass, center-of-mass, friction, softness, and shape, as well as the initial grasp inside the gripper and dynamics, including moment of inertia and aerodynamics. The task becomes even more formidable when dealing with non-rigid objects, which are very common in our daily human life. An accurate control and manipulation of objects are crucial elements for achieving desired outcomes in throwing tasks.

The throwing manipulation problem has gained attention over the past decade, and addressed by providing analytical and supervised learning-based approaches [2], [5]. However, the existing research primarily focuses on underhand throwing tasks of rigid objects and to the best of the author's knowledge, no studies have explored the realm of overhand throwing tasks. Additionally, the field lacks comprehensive efforts in harnessing the potential of contact-rich tactile perception and reinforcement learning (RL) methodologies to enhance the performance of throwing tasks. Motivated by these gaps, this work investigates challenges and solutions for successful overhand throws of nonrigid objects. Recent work on high-resolution tactile feedback has provided intuition to estimate the object-centric properties by performing in-hand manipulation [6], which are critical to determine the object's behavior for throwing. It poses challenges to develop a dynamics model for unstructured environments, such as throwing a volleyball or a dart (both involves overhand throw motion) or transferring an object for packaging in a

warehouse, while learning dynamics from raw data is an easier task on the other hand.

Robot learning for an object transfer via throwing manipulation is challenging for a simulator because simulators lack the accuracy to represent the intricacies of real-world scenarios characterized by high levels of noise and randomness. Particularly, in the case of a non-rigid object hitting a far target location after undergoing in-hand manipulation between the fingers equipped with tactile sensors, it becomes imperative to address the challenge posed by substantial noise within the system. There is always a trade-off when transferring learned controlled tasks to real robots.

Herein, we explore a more complex problem: Can a real robot arm learn to *throw unknown non-rigid cylindrical objects under the influence of moment of inertia (allowed by in-hand manipulation) to a target location by overhand throw with a single policy*. Our goal is to develop a framework to learn target-oriented object throwing manipulation with contact rich active tactile exploration. A novel **TT-RL** (Tactile Throw Reinforcement Learning) policy framework which we called *DartBot* is proposed for learning robotic skills to throw arbitrary non-rigid cylindrical objects.

In this work, our focus is solely on utilizing tactile feedback to incorporate object-centric information for non-rigid objects into the learning process, without relying on vision-based methods. While vision-based approaches have demonstrated promising performance in learning physical object representations through dynamic interaction for various manipulation tasks [7], their applicability remains limited to structured environments. Furthermore, these methods do not adequately address the challenges associated with deploying deep learning based vision systems in real world scenarios. As an attractive alternative, tactile sensing emerges as a viable option for addressing these limitations and providing valuable insights for object-centric learning.

The environment we selected for the investigation of this work consists of a magnetic dart board fixed on a wall and various objects that exhibit the characteristics of a ball and shaft system, comprising a rigid head (ball) with a magnetic face and a deformable cylindrical shaft. Specifically, we choose a magnetic dart as template object for this study. We approach the throwing manipulation strategy by first grasping a dart object (from the shaft), tilting it to different angles to acquire knowledge about its physical properties using tactile feedback. The RL agent then deduce proper action parameters from the learned policy to regrasp the object and executing an overhand throw motion to transfer the object towards the dartboard while the object is undergoing spin motion in the air (due to its moment of inertia) before establishing contact and sticking to the target location.

We evaluated the performance of our proposed method using both seen and unseen objects. These objects are created using a modular system, allowing us to vary their key parameters such as length, mass, and center-of-mass. Through comprehensive evaluation, our method demonstrated superior performance in throwing both seen and unseen dart objects, achieving a success rate of 88.75% with a target hitting mean

error of 2.20cm for seen objects, and a success rate of 95% with a target hitting mean error of 3.15cm for unseen objects.

To the author's knowledge, this study is the first of its kind addressing the high-speed overhand throwing manipulation problem on a real-robot by integrating the capabilities of tactile sensing and reinforcement learning. We summarized the contributions of this work as follows:

- An RL framework, we name it **TT-RL**, to learn robotic throwing skills through the utilization of high-resolution tactile feedback.
- A method to transfer small nonrigid objects to a far distant location outside the robot workspace via overhand throwing motion.
- Our experimental findings demonstrates the learned policy's exceptional accuracy in object throwing, with mean errors of 2.20cm and 3.15cm for seen and unseen objects, respectively.

After highlighting literature in Sec. II, we present preliminaries and our approach in detail in Sec. III. Sec. IV presents training the proposed RL agent and testing of learned policy with results and discussion. Lastly, conclusion of this study and future works are given in Sec. V.

II. RELATED WORK

In this section, we provide an extensive examination of the research done in the domain of robotic throwing skills and manipulation with tactile sensing. We aim to highlight the significant contributions made in these fields by the robotic community. By examining the existing literature, we provide a thorough summary of the key studies that have paved the way for the development of target-oriented throwing using classical approaches, machine learning techniques and the integration of tactile sensing for enhanced manipulation capabilities. Through this review, we aim to shed light on the state-of-the-art approaches and identify research gaps that motivate our current work.

A. Analytical Methods for Throwing Manipulation

In the beginning, robots were provided with throwing skills by employing hand-coded or analytical analysis approaches, followed by the optimization of control parameters to plan and execute a projectile motion for an object to transfer to a desired location, a ball for example. These analytical methods aimed to understand the dynamics of the throwing motion and refine control strategies for achieving accurate and targeted throws. However, modeling the dynamics of such unstructured environment pose challenges because it requires precise information about the object-centric properties, gripper and environment boundary conditions, limiting its applicability to real-world scenarios [8]. Throwing a ball at high speed in [9] is demonstrated with robotic setup having one rigid link and one flexible link to derive an analytical approach using Hamilton's principle. The performance is evaluated using a ball with known mass and moment-of-interia. In [10], a stereo vision system is utilized to calculate a cubic polynomial-based ball throwing transformation, by which the robot

system effectively throws the ball towards a visually tracked target. The studies [11] and [12] showcased the capabilities of a single degree of freedom (1 d.o.f) robot in controlling three kinematic variables for the task of throwing a disk (ball). In another work [13], an analytical method is presented for generating trajectories to anticipate the magnitude, direction, and duration of the end-effector velocity during an underhand throwing task performed by a humanoid robot. Parts sorting and assembly task was accomplished in [14], by throwing motion planning and optimizing control parameters. The work presented in [15] introduced an approach based on a kinetic chain for achieving high-speed swing motion, with a particular emphasis on the effective transmission of torque. While in another work [16], a mathematical throwing model was devised for ball throwing utilizing a one joint robot. The work [17], optimized the end-effector shape for planar object throwing. In [18], the authors put forward a control strategy that incorporates a high-response end effector to mitigate the delay arising for a low-response robot arm during ball throwing. Whereas the work [19], proposed a numerical framework designed to optimize throwing motion for a humanoid robot. This framework not only enables the robot to accomplish sidearm and maximum distance throwing tasks but also generates dynamic features to enhance its performance. Moreover [20] designed a latching mechanism based gripper for grasping and throwing. Such methods have made valuable contributions to the development of robotic throwing manipulation. However, these methods possess limitations that impede their effectiveness and practicality in real-world scenarios, such as the challenge of generalizing their performance across changing dynamics and diverse objects.

B. Learning Based Throwing Manipulation

Numerous investigations have delved into machine learning-based approaches for throwing manipulation tasks, showcasing the superiority of robots' performance compared to traditional analytical methods. A recent study [2] focused on training a robot to perform rapid grasping and throwing of rigid objects towards a specified target location. The approach involved learning control parameters to grasp and throw the object through trial and error, utilizing visual observations. By incorporating deep networks, they utilized predictive capabilities to estimate residuals alongside control parameters acquired from a physics simulator. Therefore, the approach relies on prior knowledge of throw physics and necessitates a substantial number of real-world samples. Another study [21], aimed to develop an adaptive search algorithm for assessing the feasibility of a parameterized throw trajectory. In [22], efforts were made to reduce the computational costs and develop a machine learning-based model for object dynamics and robot kinematic feasibility. This approach enabled a mobile manipulator to rapidly generate throwing motions within 1 ms for rigid objects, considering their flying dynamics and specific target positions. In [23], an iterative residual policy framework was proposed for manipulating deformable objects and was demonstrated with hitting a target with a whipped rope and reaching a target with a

swinging cloth, highlighting its generalization capabilities to real-world dynamics. Furthermore in [24], an open-loop controller was implemented for a soft arm by utilizing an artificial neural network that approximates the correlation between the actuation attributes and the desired landing location.

RL algorithms, on the other hand, offer an alternative by learning accurate throwing policies without prior knowledge. Early work utilized a deep neural network to map image observations to sequences of motor activations, which facilitated the learning of predictive policies for skilled robot tasks. [25]. The method was showcased through the application of a ball throwing scenario. Another study demonstrated the learning of a hitting skill, specifically table tennis, through an imitation and RL combination [4]. This acquired skill was then extended to develop a skill for catching. In [5], used Decision Transformers (DT) for transferring a simulation-based throwing policy to the real environment, achieving throws with minimal training data from the real-world. In another study [26], a hierarchical framework was introduced that utilized deep RL to empower a quadrupedal robot with the ability to execute accurate shooting maneuvers for a soccer ball towards randomly placed targets in real-world scenarios. Considering obstacles in the environment [27], proposed a RL approach to transfer objects by throwing into a moving basket while navigating obstacles in the path. In another work [28], deep RL was employed for a soft robot to generate accurate throwing motions for round objects. Although notable advancements achieved in the domain of learning throwing skills using vision, the incorporation of local grasp information of the grasped object, which greatly impacts successful throwing tasks, has been lacking. None of these studies have specifically explored the integration of feedback on grasped objects, such as tactile sensing. In contrast we focus on integrating high-resolution tactile feedback into the learning process of a RL agent, and show that such local adjustments are useful to enhance the robustness of throwing skills.

C. Tactile Sensing for Enhanced Manipulation

Tactile sensing, which involves the ability to sense and interpret the physical contact with object/environment, plays a crucial role in enhancing robotic manipulation skills [29]. The inclusion of tactile sensing can significantly improve the aforementioned robot throwing abilities. Recent studies on high-resolution vision-based tactile sensing [30], [31], [32], [33] have facilitated the seamless integration of real-time tactile feedback into deep learning solutions. In [6], the study focused on learning the object's physical properties through tactile exploration and utilized this information to predict angles for dynamic swing-up manipulation of unseen rigid objects. In addition, the quality of a grasp is crucial for enabling robots to effectively handle uncertainties in the environment. In the work [34], presented a regrasp control policy that utilized tactile sensing and a deep convolutional neural network to improve the quality of grasps through local adjustments, resulting in a significant increase in the success

rate of grasp actions on various objects. In [35], a tactile-based framework was developed for learning regrasping behaviors, which improved grasp success rates from 42 percent to up to 97 percent through accurate grasp outcome prediction and subsequent regrasping in case of failure. In another work [36], a tactile-based approach using an action-conditional framework was developed to enhance robotic grasping by incorporating visuo-tactile data, resulting in improved grasp adjustment outcomes. Such studies have inspired our current work to incorporate high resolution tactile feedback in the learning loop of high-speed dynamic throwing manipulation.

III. APPROACH

The aim of DartBot is to make a robot arm capable of throwing arbitrary nonrigid objects, under the influence of moment of inertia, to a target location with a single subsequent exploratory action. We present a real-time approach for a robot to learn throwing skills solely based on tactile sensing, illustrated in Fig. 1. This section provides an overview of the RL algorithm we employed, followed by the problem formulation and the proposed TT-RL framework.

A. Preliminaries

MDP: An Markov Decision Process (MDP) encompasses four essential elements: $(s_t, a_t, p(s_{t+1}|s_t, a_t), r(s_{t+1}|s_t, a_t))$. In this formulation, s_t represents the continuous state (of the environment) while a_t represents action taken at time step t . The transition probability function $p(s_{t+1}|s_t, a_t)$ characterizes the likelihood of transitioning to s_{t+1} (next state) given a_t and s_t . Additionally, the term $r(s_{t+1}|s_t, a_t)$ represents the collected reward followed by a transition in state.

TD3: An off-policy reinforcement learning algorithm known as Twin Delayed Deep Deterministic Policy Gradient (TD3) [37] is suitable for environments that feature continuous action spaces. TD3 is an actor-critic method that simultaneously learns two Q-functions, Q_{ϕ_1} and Q_{ϕ_2} , through minimizing the square Bellman error. A single target is used for both Q-functions, determined by selecting the smaller target value:

$$y(r, s', d) = r + \gamma(1 - d) \min_{i=1,2} Q_{\phi_i, \text{targ}}(s', a'(s')) \quad (1)$$

Here, r represents the reward value, s' is the next state, d represents whether state s' is terminal, γ is the discount factor (0.99 in our case), and a' denotes the next action. Subsequently, both Q_{ϕ_1} and Q_{ϕ_2} are learned by regressing towards this target:

$$L(\phi_1, \mathcal{D}) = \mathbb{E}_{(s, a, r, s', d) \sim \mathcal{D}} [(Q_{\phi_1}(s, a) - y(r, s', d))^2] \quad (2)$$

$$L(\phi_2, \mathcal{D}) = \mathbb{E}_{(s, a, r, s', d) \sim \mathcal{D}} [(Q_{\phi_2}(s, a) - y(r, s', d))^2] \quad (3)$$

Lastly, the policy is learned through the maximization of Q_{ϕ_1} :

$$\max_{\theta} \mathbb{E}_{s \sim \mathcal{D}} [Q_{\phi_1}(s, \mu_{\theta}(s))] \quad (4)$$

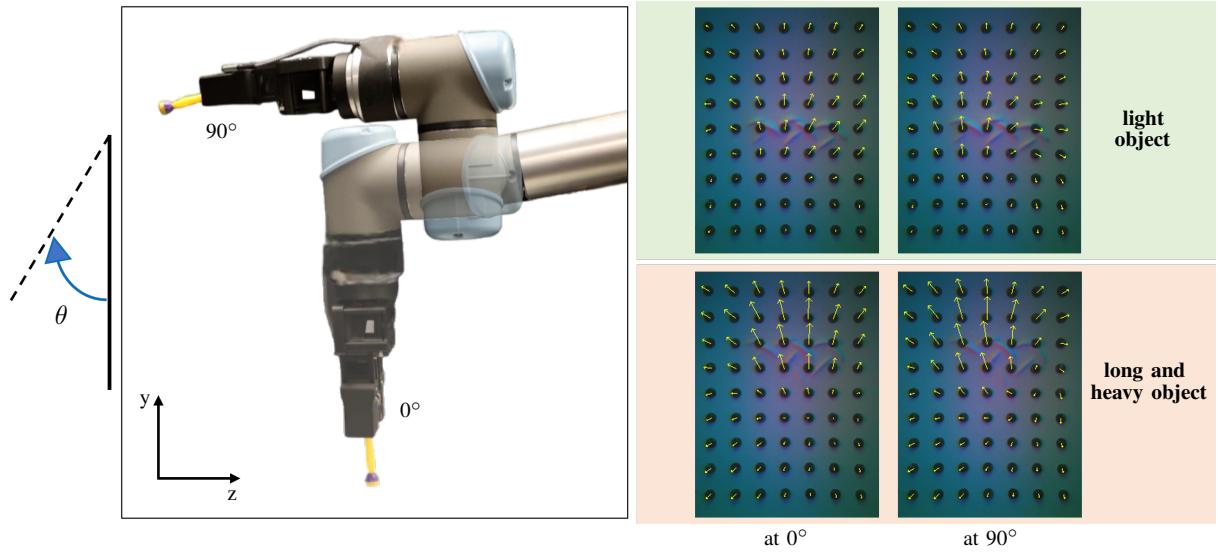


Fig. 2: Observation space for the TT-RL framework includes tactile images with marker displacements at 0° and 90° tilt motion. When grasping and maintaining objects at 0° , both light and heavy, long objects exhibit significant variation in marker displacement values. Similar variations occur when tilting these objects to 90° . The feature vector $obs \in \mathbb{R}^{24}$ consists of the 20 marker displacement values and 4 action space parameters of the previous timestep. The 20 marker displacement values are obtained by concatenating the top 10 values (from 63 available marker displacements) at each grasp pose (0° and 90°) which are arranged in ascending order by magnitude.

Here, D represents a transitions set (s, a, r, s', d) , and μ_θ is the target policy.

B. Problem Description

Given the pose of a non-rigid cylindrical object and the goal location, the overhand throwing task is constructed as the problem of finding optimal parameters: to regrasp the object at suitable posture, to plan the throw trajectory and the release time, for a successful throw motion to hit the target location. The task at hand can be represented using a fully observable Markov Decision Process (MDP), and it can be effectively solved using an off-policy reinforcement learning framework. In the subsequent subsections, we delve into the comprehensive formulation of the throwing task within the RL framework.

1) Observations: In the RL environment, obtaining task-related observations plays a crucial role in enabling the agent to learn appropriate actions. As previously emphasized, our focus lies in integrating tactile sensing into the learning process of the throwing task. We leverage GelSight tactile sensor technology [38] in this study, which offers force tracking by pixel displacement [29] and 3D reconstruction of the geometry for grasped object [30]. The sensor offers a grid of 9×7 sparse markers on its tactile skin. A feature vector is employed to define continuous observations, encompassing contact-rich tactile feedback once the object is grasped. Notably, we explored that rotating the grasp to different angles yields significant tactile data, facilitated by a vision-based tactile sensor with markers on its sensing skin.

Specifically, at each time step i , we grasp the object and raise it to a specific height along the y -axis of the gripper's frame while gripper is at an angle of 0° relative to the x -axis, as depicted in Fig. 2. Then, we tilt the gripper to 90° relative

to the gripper's x -axis, and record the tactile marker flow at both instances. Among the 63 available markers, we select the top 10 markers (at each instance) based on their deflection magnitude. These selected 10 marker values at each gripper pose (0° and 90°) are concatenated to obtain an array of 20 values, and are represented as $obs = (m_1, m_2, \dots, m_{20}) \in \mathbb{R}^{20}$. Through our explorations, depicted in Fig. 2, we have determined that the deflection of these 10 markers adequately captures essential information sufficient to embed physical properties of the deformable object, such as its mass and center of mass. There are two primary reasons for using only 10 marker displacement values instead of all 63 at each pose of 0° and 90° . Firstly, we grasp the object at its deformable shaft end with a force that allows in-hand rotation during the motion trajectory before release. This causes the object to make contact only with the edge of the shaft end, which has a larger diameter compared to the remaining part of the shaft. As this diameter is consistent, different darts will have a very similar size contact area. Secondly, the object is held in a fixture to maintain its straight orientation, although slight variations in grasp posture may occur due to the mesh structure, which can be considered negligible. These two assumptions led us to focus on 10 marker displacement values instead of utilizing all 63. By using these 10 markers, we are able to extract the significant information required for this task and allow for greater generalization due to reduced dimensionality. Furthermore, we have incorporated the four action space parameters from the previous timestep into the observation space, resulting in a fully observable MDP. As a result, the final $obs \in \mathbb{R}^{24}$.

2) Actions: We define the action space as a vector of four elements. This four dimensional continuous action space includes angle of robot arm base joint (ψ), grasped object height (h_o) relative to the sensor frame, grasped object

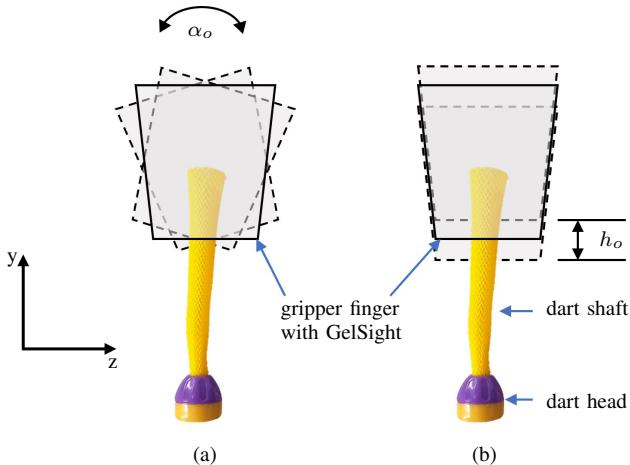


Fig. 3: Action space parameters for the TT-RL framework: The two parameters (from $a \in \mathbb{R}^4$) that the agent learns to effectively regrasp the object for a successful throw, (a) the orientation of grasped object relative to the sensor frame, α_o , achieved by rotating the gripper about the x-axis, and (b) the height of the grasped object h_o relative to the sensor frame achieved by translating the gripper along y-axis.

orientation (α_o) relative to the sensor frame as illustrated in Fig. 3, and the object release time (t) for throw. The allowable ranges for the proposed action space are given in Table I, which are obtained empirically and briefly discussed later in Sec. IV, to transfer the object within the target region while ensuring environment safety. Further, the angle ψ is empirically employed to generate the throw trajectory, enabling the object to be hit at various locations. Instead of requiring the method to learn all six joint angle values of the robot arm for throw trajectory, we divided the trajectory into seven points in the robot's joint space. The agent learns the base joint angle ψ , while the remaining points are predefined, reducing the complexity of the approach. More detail is provided in Sec. IV.

Furthermore, the gripper's opening width is empirically controlled to facilitate in-hand manipulation at two key stages of the overhand throwing task and the subsequent learning process. Initially, a tight grasp is maintained to ensure that the object is securely held while the robot arm moves from the object pickup location to its initial pose on the throwing trajectory, while consistently preserving the generated action parameters, h_o and α_o , of the agent. Once the robot arm reaches the desired initial pose configuration, the gripper's opening width is adjusted to enable controlled rotation (in-hand manipulation) of the object within the gripper fingers, as illustrated in Fig. 4. This adjustment and in-hand manipulation of the object are crucial for the subsequent phase of the overhand throwing task, namely, the execution of the throwing motion trajectory. Additionally, in-hand manipulation of the object occurs throughout the throwing motion trajectory prior to release, which we refer to as pre-release in-hand manipulation. This manipulation contributes to the generation of the object's moment of inertia, thereby influencing the mid-air dynamics of the object and its landing within the target region.

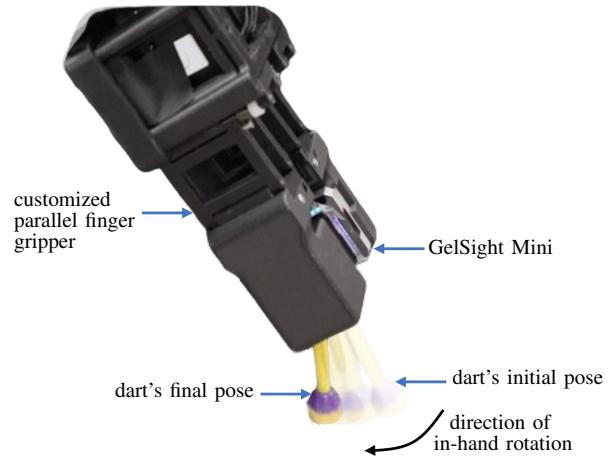


Fig. 4: In-hand manipulation of the dart object: The dart object is firmly held between the two-fingered parallel gripper, which is equipped with a GelSight Mini tactile sensor. The robot maintains this firm grasp on the dart from the pickup location along the way towards its initial pose on the overhand throwing trajectory. Upon reaching the designated pose, the gripper's opening width is adjusted empirically, enabling in-hand manipulation of the dart. As a result of gravity, the dart rotates from right to left during this process.

3) Rewards: A throw is considered successful if the object's magnetic face lands on the dartboard, so that it is attached on the board, after undefined spins in the air as shown in Fig. 6. The reward value is calculated based on its distance from the target location. The target to hit lies on a disc having radius of 15cm. By default, the target location is set as $x_g = 0$ and $y_g = 0$ i.e., the disk center, and a value of 150 is assigned to this location. By processing difference images captured by the Microsoft Azure Kinect camera, a reward value is computed based on the linear distance between the actual landing location of the object and the target landing location. A high negative penalty of -10 is rewarded in case the object fails to reach/land within the 15cm radius circle.

C. Tactile Sensing for Physical Embeddings

We humans, granted by enormous sensing capabilities throw objects by feeling the grasp strength, grasp location, adjusting the grip and deciding the arm pose when to release a grasped object to hit a target location. Achieving similar skills for a rigid body robot is difficult on the other hand. Over the past decade, as highlighted previously in Sec. II, research on high resolution vision based tactile sensing provides promising performance by equipping robots with

TABLE I: Randomization range of action space parameters

Parameter	Symbol	Range	Unit
Robot base joint angle	ψ	[-2.0 2.0]	degree
Grasped object height (w.r.t sensor frame)	h_o	[-1.0 1.0]	mm
Grasped object orientation (w.r.t sensor frame)	α_o	[0.0 3.0]	degree
Object release time	t	[0.76 0.84]	sec

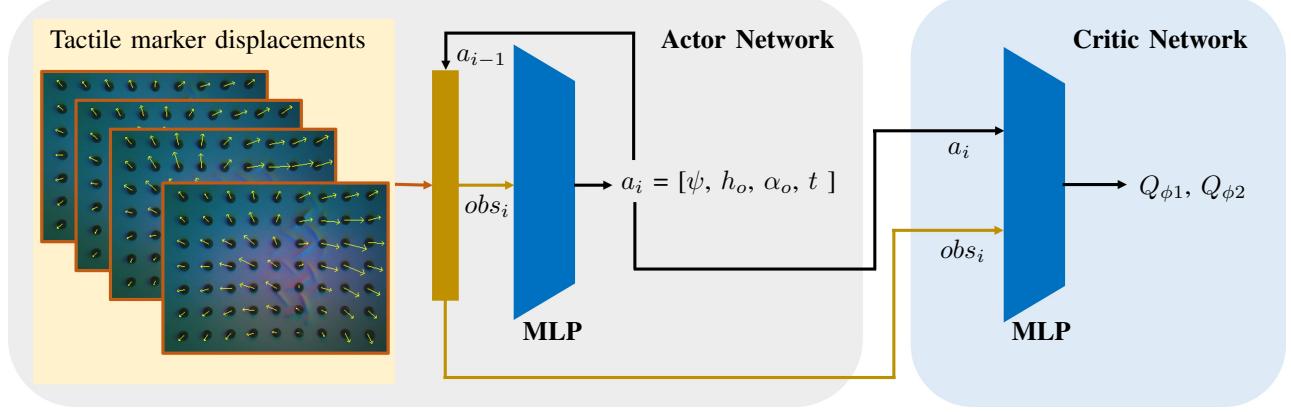


Fig. 5: The Actor and Critic neural network models in Twin Delayed Deep Deterministic Policy Gradient (TD3). The Multilayer Perceptron (MLP) consists of two fully connected layers with 256 hidden units each for both the networks. The actor network takes a 24×1 obs vector that consist of tactile features obtained after performing two tilt motions and the action space parameters of the previous timestep as input and produces a list of action space elements as output. The critic network takes the obs vector and action space as inputs and outputs learned policies Q_{ϕ_1} and Q_{ϕ_2} .

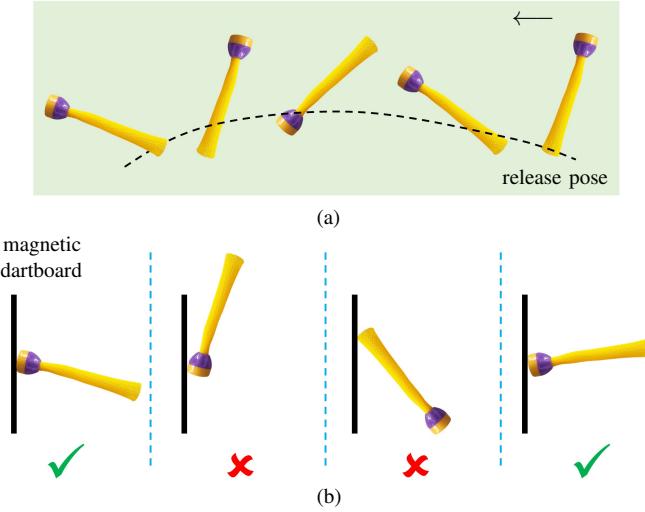


Fig. 6: The impact of moment-of-inertia resulting from pre-release in-hand manipulation. (a) illustrates the spin motion of the non-rigid object in the air after release from the gripper, with the black dotted curve representing its travel trajectory from right to left. (b) showcases different hitting scenarios on the target (magnetic dartboard), with successful hits and attachments indicated by green ticks and failure cases denoted by red crosses, where the magnetic face is not facing towards the dartboard.

real time contact rich feedback [38] on their interaction with the environment. The underlying concept here is that humans, through their interaction with objects using their hands over a certain duration, can develop an instinctive insight of the grasped object's dynamics. This instinctive information enables them to make more accurate estimations for the physical attributes of the object, mass for example.

We choose a GelSight Mini [39] sensor that contains a grid of 9×7 markers to provide real time tactile feedback. We employed the flow of these markers upon grasping the desired object to throw. We propose to encode this tactile information as pseudo embedding to estimate object-centric properties and treated as the observation space as explained in Sec. III-B1 for the RL agent. In contrast to the method employed

in [6] which utilized a sequence of tactile images to learn the physical properties of the objects, our approach focuses on utilizing solely the numerical data of marker displacement and orientation. These measurements are obtained by raising the grasp object to a specific height along gripper's y-axis and then tilting the grasp to 90° about the gripper's x-axis, as depicted in Fig. 2. By leveraging this reduced-dimensional tactile feedback, we are able to infer valuable information that allows us to differentiate between object physical properties. This approach significantly decreases the computational cost associated with processing high-dimensional tactile images.

D. Deep RL Framework

To train the TT-RL policy with the continuous action and observation spaces, we choose the Twin Delayed Deep Deterministic Policy Gradient (TD3) [37], via stable baseline3 [40]. TD3 is an actor-critic method which concurrently learns two Q-functions, Q_{ϕ_1} and Q_{ϕ_2} , by means of square Bellman error minimization as explained in Sec. III-A. TD3 outperforms the tasks that requires fine-grained control as it is designed to handle continuous action spaces. Based on our findings, tactile exploration during the throwing manipulation task often involves encountering noisy feedback. Fortunately, TD3 demonstrates its capability to effectively handle and accommodate these noisy feedback signals.

To construct the actor network, we utilize a feature vector obtained from GelSight feedback and the action parameters of previous timestep alongwith Multilayer Perceptron (MLP) model, as depicted in Fig. 5. The inputs to this network are tactile marker displacements and the previous timestep action values while the outputs correspond to the variables within the action space. On the other hand, for the critic network, we employ an MLP model illustrated in Fig. 5. This model is responsible for learning the Q_{ϕ_1} and Q_{ϕ_2} functions by taking observation space and action space variables as inputs. The MLP model for both the networks consists of two fully connected layers with 256 hidden units. The complete TT-RL framework for our problem is developed within the OpenAI GYM environment [41].

E. Objects and Dataset

Ensuring the generalization capability of our model to previously unseen objects is crucial, and it heavily relies on the diversity of training conditions. To address this challenge, we have devised a modular system inspired by [6] to efficiently create a comprehensive set of template objects by altering their length, mass, and center of mass. Our template objects are depicted in Fig. 8. Each object consists of three main components: a rigid head, a magnetic face, and a non-rigid shaft with a mesh structure. The gripper grasp the object by interacting with the mesh shaft. To modify the length and mass of the objects, we employed 3D printed plastic discs having thickness of 2mm and 3mm. Additionally, to vary the mass while maintaining the length, we incorporated various combinations of small steel balls, each weighing 0.11 grams. Through this approach, we created five distinct objects with mass of 3.81gm, 4.39gm, 4.06gm, 4.14gm, 4.53gm and their respective lengths of 78.66mm, 82.01mm, 80.00mm, 82.00mm, 83.04mm. Out of these five objects, we use two objects to train the TT-RL policy, while the other three objects (shown in blue dashed box in Fig. 8), which are unseen to the agent, are used to evaluate the performance of the trained policy.

F. Approach for Object Releasing

Initially, our proposal involves integrating tactile feedback data into the TT-RL agent to facilitate learning of the object dynamics both during pre-release in-hand manipulation and during the undefined spins in the air. This can be achieved by tracking the object’s pose throughout the throwing trajectory motion with respect to the sensor frame. We explore a PCA-based approach, as demonstrated in [29], to encode this pose information. Specifically, we focus on the object’s pose at the moment the gripper opens to release it, considering that in-hand manipulation occurs during the throwing trajectory due to robot velocity and trajectory shape. We have observed that the object release pose information plays a crucial role as it directly influences the behaviour of the object after release. Consequently, it has impact on the throwing performance and the success rate of hitting the target.

To enhance the agent’s understanding and control of the object’s moment of inertia after release, we aim to incorporate this pose information as a action space parameter. However, the nonrigid and mesh structure of the object’s shaft present challenges in obtaining the pose information using the PCA-based method. As a result, we devised a solution by incorporating the object release time (t) into the TT-RL framework, which yielded significant improvements in model convergence.

We aim that the policy learned by above TT-RL framework with training on real robot arm equipped with GelSight Mini sensor, will outperform to throw unknown nonrigid cylindrical objects towards a target location with high precision and accuracy.

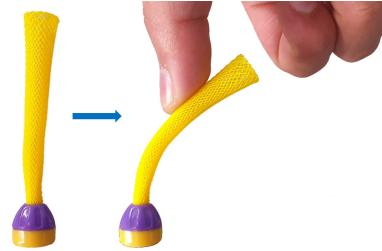


Fig. 7: Nonrigid template object. The applied force causes the mesh shaft structure to bend.

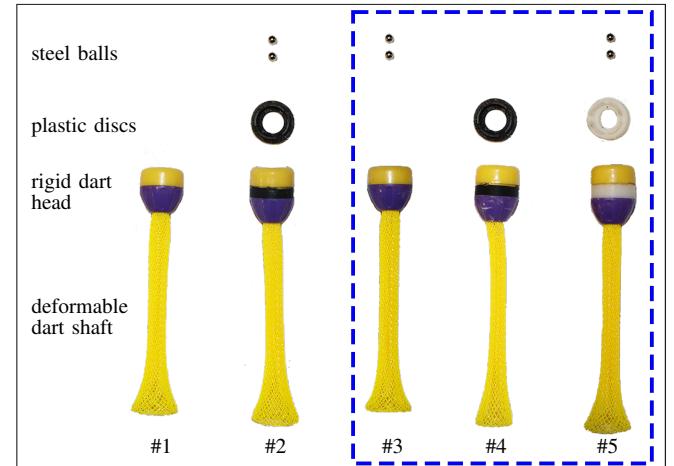


Fig. 8: For our experiments, we utilized template objects, as depicted in the bottom row, composed of three primary components. These components include a rigid head distinguished by a purple region, a magnetic face consisting of a magnet inserted into the top yellow part, and a non-rigid shaft with a yellow color and mesh structure. To generate a diverse range of objects with distinct physical properties, we employed small steel balls showcased in the top row, along with 3D printed plastic discs featured in the middle row where black disc has thickness of 2mm (object#2 and #4) and white disc has thickness of 3mm (object#5). The objects within the blue dashed box are unseen to the agent and are used for testing the performance of learned policy.

IV. EXPERIMENTS AND RESULTS

Here, we present a comprehensive description of the hardware setup utilized to investigate, train, and test the TT-RL framework for the overhand throwing manipulation task. We outline the key components and configurations of the hardware system that enabled us to conduct comprehensive experiments and evaluate the effectiveness of our approach. Particularly, we assess the performance of the proposed framework using two primary metrics: the success rate of overhand throwing task, defined as the proportion of times the thrown object lands on the dartboard after undergoing undefined spins in the air, and the distance from the target landing location. More specifically, our investigation aimed to determine whether our proposed method can effectively learn to throw arbitrary deformable objects towards a target location, considering the challenges posed by noise and uncertainties in a real-robot environment.

TABLE II: TD3 hyper-parameters

Parameter	Value
Network Depth	2
Network Width	256
Batch size	128
Learning rate	10^{-4}
Learning Starts	300
Train Frequency	5 Episodes
Discount factor (γ)	0.99
Polyak Update	0.005
Gradient Steps	1
Action Noise	0.05 (5%)
Policy Delay	2
Target Policy Noise	0.2
Target Noise Clip	0.5
Replay buffer size	10^6
Activation function	ReLU
Target update interval	1
Seed	1

A. Experimental Setup

We trained the proposed throwing policy on the real hardware. The experimental setup we utilized is shown in Fig. 9. The hardware setting includes a UR5e robot arm, a customized adaptive two-fingered gripper [29] and a GelSight Mini sensor [39] attached at the fingertip of gripper as shown in left bottom in Fig. 1. Furthermore, it is important to note that a recycling system, depicted in Fig. 9, is employed to facilitate the training process autonomously, without the need for human involvement. This recycling system comprises a dropping four-bar linkage mechanism, a guideway, a sliding collection funnel and a reorientation cone. To ensure safety, we choose a circular magnetic dartboard with a radius of 15cm, mounted on a wall at a distance of 1.7m from the robot base, and utilized a deformable shaft magnetic dart as the template object for our experiments. A Microsoft Kinect camera is positioned above the robot arm, facing towards the dartboard. Its purpose is to detect the landing location of the thrown object, enabling measurement of the reward value. Our approach involves training the RL agent on two distinct objects and evaluating its performance on three additional objects that the agent has not encountered before. The objects are depicted in Fig. 8. One complete training cycle consists of the following sequential steps: grasping the object by its deformable shaft end while it is positioned vertically within one of the reorientation cones, raising it to a predetermined height, tilting it to a 90° angle to gather tactile measurements (as described in Sec. III-B1), releasing the object into the collection funnel to allow it to reach the reorientation cone, firmly grasping the object and positioning the robot arm to the desired initial configuration, adjusting the gripper's opening width to facilitate in-hand manipulation (as discussed in Sec. III-B2), executing the throwing motion, measuring the resulting reward, and finally returning the object to the reorientation cone for the next cycle using the recycling system.

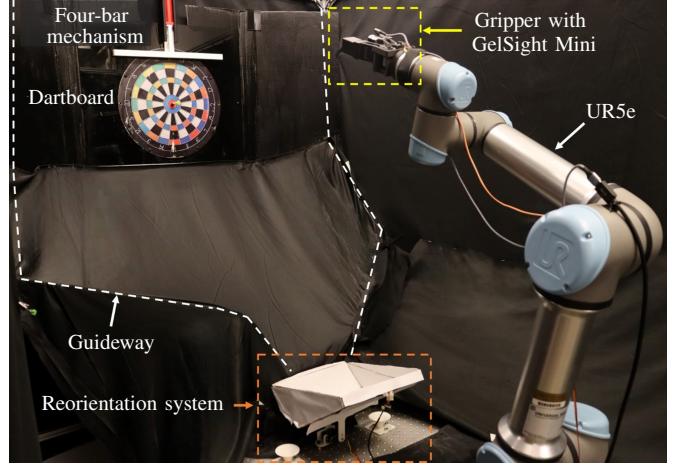


Fig. 9: The experimental setup employed for the autonomous training of the RL agent consisted of a UR5e robot arm, a customized parallel gripper equipped with a GelSight Mini sensor, and a magnetic dartboard. Furthermore, a four-bar linkage mechanism, a guideway, and a reorientation system are utilized to transfer the dart object from the dartboard to the cone for the subsequent timestep, facilitating autonomous training process without human involvement.

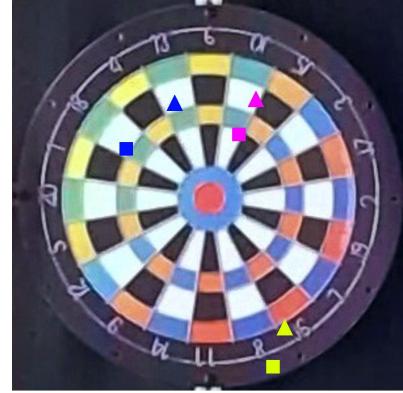


Fig. 10: Exhibiting the result of various action parameters for throwing two different dart objects. The pyramid shows hit location for dart#1 while square box indicates dart#2. The values for action space parameters are within allowable ranges. We tested three different combinations of action parameters whose hit locations are indicated by yellow, blue and purple color. We observe that throwing dart#1 & dart#2 objects with one action combination results in hitting different locations on the dartboard.

B. Training RL Agent

As briefly discussed in Sec. III-D, we employed TD3 [37] RL algorithm via stable baseline3 [40] to train the robot. The hyper-parameters of TD3 are given in Table. II. As mentioned in Sec. III-B1, the observation space of the system includes marker displacement measurements and the action values from the previous time step. To obtain the marker displacement values, we perform specific motions with the grasped object. Firstly, we raise the object to a predetermined height while maintaining a grasp angle of 0° relative to the gripper's x-axis. Then, we tilt the grasp to 90° relative to the gripper's x-axis. This sequence of motions is repeated 75 times for each object, and the resulting displacement values are recorded and averaged. The reason for this is the asymmetric deformation of the dart shaft's

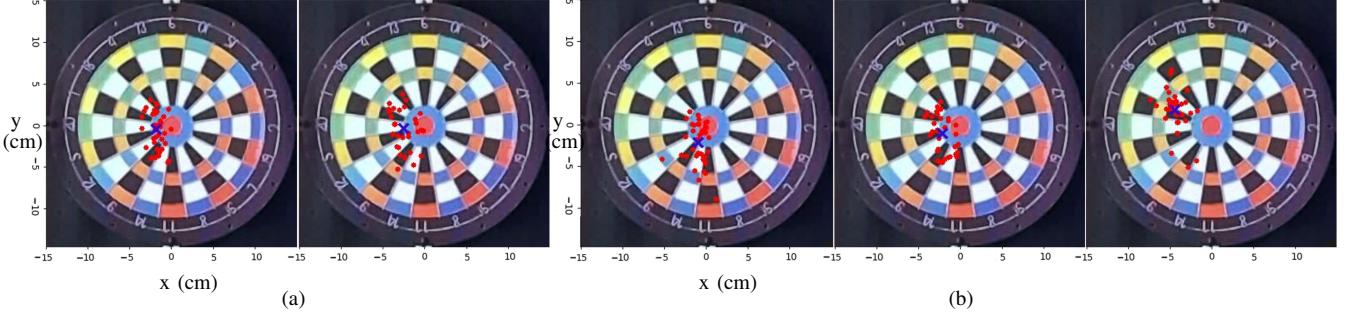


Fig. 11: Hitting locations of the test experiments marked with red dots on the dartboard, (a) seen objects used in training the RL agent, (b) objects unseen to the RL agent. A blue cross on each image shows the middle point for all successful throwing trials calculated as mean (x_m , y_m) along x and y axis, as listed in Table. III.

TABLE III: Experiment results of learned robot overhand throwing skills.

Object #	Object Head Mass (gm)	Object length (mm)	Success Rate	Mean Distance from Goal (cm)	Standard Deviation (cm)	Coordinates of Hit Locations Mid Point (cm) (x_m , y_m)
1	3.81	78.66	40/40	1.89	0.957	(-1.8, -0.5)
2	4.39	82.01	31/40	2.50	1.38	(-2.5, -0.5)
3	4.06	80.00	40/40	2.43	1.89	(-1.1, -2.2)
4	4.14	82.00	34/40	2.34	0.985	(-2.1, -1.0)
5	4.53	83.04	40/40	4.68	1.32	(-4.3, 1.8)

mesh-like structure, which occurs even when subjected to the same grasp force. By pre-recording the marker displacement values for each object and motion, we address issues such as minimizing training time on real hardware and preventing non-convergence of the model. Prior to training, we found that the marker displacements recorded after grasping the dart in the same pose exhibited a standard deviation of approximately 15%. Among 63 markers available on the tactile sensor skin we selected top 10 displacement values (at 0° and 90°) as a feature vector to represent pseudo-embedding for the object physical properties.

To ensure safety, we carefully select appropriate ranges for the parameters in the action space listed in Table. I. These ranges are chosen to prevent the robot from colliding with its environment while also ensuring that the object can reach the designated target region. To simplify the throw motion, we reduce the complexity of the 6-DOF robot arm to a 1-DOF system. We achieve this by considering the base joint angle, denoted as ψ , as the variable in the action space within the joint space. By controlling the entire robot arm in the joint space and incorporating a blend radius value of 0.25 for trajectory smoothing, we effectively reduce the system's degrees of freedom and streamline the throw motion. The velocity for each joint of the robot arm is set as 3.14 rad/sec. Further, by incorporating action noise of 5%, the ranges of the action space are designed to enable hitting all locations on the dartboard, including its surroundings. This ensures that the agent has the capability to explore and target various positions effectively during the throwing process. To ensure diversity in throwing outcomes, we have confirmed that executing the same action values for two distinct objects results in hitting different locations

on the dartboard, illustrated in Fig. 10. Considering all the aforementioned factors, we conducted training on the real-robot world for a total of 1254 steps using two distinct objects. This training duration amounted to approximately 20 hours, utilizing an Intel Core i7 processor with 64GB RAM and an Nvidia GeForce RTX3080 graphics card with 10GB dedicated RAM. We set the maximum allowed timesteps for each episode as 10. Each episode is terminated if meeting any of the following conditions after 3 timesteps: i) the rewards reaches a value of 135 or higher, ii) the reward value is not higher than 20, iii) the change in reward value is 20. We choose these conditions to achieve early model convergence on the real hardware training setup.

C. Results and Discussion

The fundamental objective of this study is to uncover the underlying physics involved in throwing a deformable cylindrical object, specifically a dart, which undergoes undefined spins in the air following pre-release in-hand manipulation. The dynamics of the object in mid-air can be attributed to two main factors: the gripper's intrinsic dexterity, which in our case called as pre-release in-hand manipulation, and the dynamic extrinsic dexterity of the object arising from its generated moment-of-inertia, as briefly described in Sec. III-B2.

For the learned throwing policy, we evaluated the throwing performance by conducting the throwing experiments using two sets of objects: a "seen objects" set consisting of two dart objects used during training, and an "unseen objects" set consisting of three dart objects not encountered during training. Each object was thrown 40 times. The throwing success rate, which measures the successful landing on the

dartboard (which can only occur if the dart's magnetic face is towards the dartboard as shown in Fig. 6(b), the mean distance from the goal location (center of dartboard), and the middle point for all the successful throwing trials for each object, respectively, are presented in Table. III. Furthermore, the objective of the employed RL agent is not solely focused on hitting the dart at a specific target. Instead, it aims to learn how to enable the dart to land within an acceptable range of position and orientation, relative to the dartboard surface, after undergoing mid-air dynamics. The success of the overhand dart throw is determined by whether the landing position and orientation fall within the desired range, which facilitates its attachment to the dartboard and is rewarded accordingly.

1) Advantage of Pre-release In-hand Manipulation: To validate the hypothesis put forward in Section III-B2 regarding the importance of pre-release in-hand manipulation in the overhand throwing task involving deformable objects, we conducted a series of experiments. First, we securely grasped the dart at its shaft, positioned vertically from the reorientation cone and positioned the robot arm at the desired initial pose on the overhand throwing trajectory. Subsequently, we executed the overhand throwing motion while maintaining a firm grasp on the dart, thus preventing any in-hand manipulation. This throw attempt resulted in the object falling far from the dartboard (specifically on the guideway), following a low-trajectory projectile motion without any mid-air spins. Without in-hand manipulation, the dart maintained a fixed orientation relative to the sensor frame throughout the robot throwing motion. This lack of dynamic adjustments in the object's orientation and position resulted in a less optimal trajectory and reduced control over the dart's flight path. As a consequence, the dart fell far from the intended target, failing to achieve the desired outcome i.e., to hit the dartboard. Hence, the obtained success rate for this set of experiments was 0 out of 10 attempts.

On the other hand, the importance of pre-release in-hand manipulation becomes evident when considering the success rates presented in Table III for both the seen and unseen sets of objects. These success rates reflect the number of instances, out of the total 40 throws, in which the dart lands within the acceptable range of orientation and position relative to the dartboard surface, thereby enabling successful attachment. We achieved 100% success rate for throwing object#1 and 77.5% success rate for throwing object#2 (seen objects). And for the unseen objects #3, #4, #5, we achieved success rates of 100%, 85%, and 100% respectively. These high success rates achieved for the case of pre-release in-hand manipulation of the dart can be attributed to two key factors. Firstly, the in-hand manipulation allows for the generation of a moment-of-inertia similar to a ball and shaft system. This moment-of-inertia influences the rotational dynamics of the dart during its flight, contributing to variations in its position and orientation relative to the dartboard surface. These variations increase the chances of successful attachment by enabling the dart to land within an acceptable range of position and orientation. Secondly, the RL agent we employed learns the optimal release time for

the dart. Through reinforcement learning, the agent discovers the precise moment to release the dart during the overhand throwing motion. This learned release time allows for optimal launch conditions, including the angle and velocity, resulting in a high trajectory projectile motion. The combination of the learned release time and in-hand manipulation provides fine-tuned control over the dart's flight, enabling it to cover a greater distance while maintaining the desired orientation and position for successful attachment to the dartboard.

2) Learned Throwing Performance: The learned overhand throwing policy exhibited superior performance, as illustrated in Fig. 11 and listed in Table. III in throwing both the seen and unseen objects. It achieved a 100% success rate for throwing object#1 and 77.5% success rate for throwing object#2 (seen objects). For the unseen objects #3, #4, #5, the agent achieved success rates of 100%, 85%, and 100% respectively. These results demonstrate the agent's ability to learn and optimize the object's launch conditions, including angle and velocity, based on the release time parameter of the action space. And the in-hand manipulation, which occurs at two key stages, plays a critical role in achieving the optimal launch conditions and the resulting mid-air moment-of-inertia of the dart. The first stage occurs at the robot's start pose of the overhand throwing trajectory (depicted in Fig. 4, and the second stage occurs throughout the robot's overhand throwing motion. This demonstrates that the agent has learned the object dynamics, including the generated moment-of-inertia, even without using the release pose information and relying solely on the release time parameter. Moreover, the agent's ability to learn and adapt its throwing strategy based on observed outcomes and variations in object properties allows for reliable and precise attachment of the dart. This is supported by the high success rates achieved even with unseen objects and the agent's adaptability to different physical parameters, which the agent discovered with the aid of high-resolution tactile feedback. The agent leverages this tactile feedback to gather detailed information about the object's properties, enabling it to adjust its throwing strategy accordingly and optimize the launch conditions for accurate throws.

The next attribute we analyze to report the performance of our approach is the mean error distance of dart hits from the goal location. We examined the deviation of hit locations from the target for each of the template objects. Notably, the agent demonstrated exceptional performance in throwing object #3 (an unseen object), as depicted in the first left image of Fig. 11(b). Out of 40 successful trials, 9 hits landed within the inner red circle of the dartboard, indicating the agent's ability to adjust the launch conditions based on observed outcomes and adapt to varying object properties.

The mean error distance increased as variations in the object's physical parameters became larger. In particular, object #5 exhibited a significant increase in mean error, measuring 4.68cm. This increase was due to notable variations in the shaft length and mass of the dart head. However, despite the increased mean error, the system maintained a 100% success rate. This finding suggests that the agent is capable of

adjusting the launch conditions to compensate for variations in object physical parameters, enabling successful attachment even with slightly different hit locations. The variations in the mean error distance can be attributed to uncertainties in object properties. Even with a thorough understanding of the object's physical properties aided by tactile feedback, inherent variations can still exist. These variations can affect the trajectory and behavior of the thrown object, leading to deviations in the hit locations. Additionally, environmental factors such as air resistance can introduce unpredictable variations in the flight path and pose of the thrown object, which can result in deviations from the goal location. Despite the agent's learned precise control of the launch conditions, these factors can influence the outcome and contribute to variations in the mean error distance.

Furthermore, we conducted a middle point analysis to examine the positions of successful hits for each dart object. The middle points, obtained by averaging the x-coordinate and y-coordinate locations of successful hits measured in centimeters, provide insights into the agent's throwing accuracy. The middle point coordinates for each of the template objects are listed in the last column in Table. III and indicated with a blue cross in Fig. 11. Objects with small variations in physical properties exhibited middle points closer to the left-bottom region of the target location, as shown in the left image of Fig. 11(a) for object #1 and the first left image of Fig. 11(b) for object #3. For objects with intermediate variations, the middle points were located on the left side of the target location, indicating some deviation but still within an acceptable range. In contrast, objects with large variations from the training objects resulted in middle points positioned in the top-left region relative to the target location (object #5, right most in Fig. 11(b)), reflecting further deviation. The results highlight the agent's adaptability in handling a variety of objects with different physical properties and its capability to adjust launch conditions to achieve successful attachment with a high level of accuracy.

Moreover, the study revealed a consistently low average standard deviation of 1.31cm across 200 throwing trials. This finding underscores the precision and reliability of the learned overhand throwing policy. Despite variations in the object's mass, center-of-mass, and length, the agent consistently achieved accurate throws with minimal variation in hit locations. This emphasizes the robustness of our framework in achieving high accuracy and precision when throwing non-rigid objects.

Additionally, we briefly compared the robot's overhand throwing performance to that of the authors (i.e., the human subjects) given the same experiment environment. The authors grasped the deformable dart shaft using two fingers, specifically the thumb and index finger. To gain an estimation of the dart's physical properties, they maintained the grasp for a certain duration. Subsequently, an overhand throwing motion was executed to transfer the dart towards the dartboard. The human participants primarily depended on the sense of touch to perceive and adjust their grip during grasping, while they relied on visual feedback to guide their throwing actions.

The success rate achieved by the human participants was 70% for object#1, 60% for object#4, and 52.5% for object#5 in 40 trials. These results are considerably worse compared to our robot's performance of 100%, 85%, and 100%, respectively, which shows effectiveness of the learned policy.

D. Failure Cases

During our experiments, the main cause of failure trials for objects #2 and #4 was the inaccurate prediction of parameters for regrasping and releasing the object. This led to two failure cases: either the dart's shaft would hit the magnetic dartboard instead of the magnetic face upon reaching the board, or the magnetic face would make contact at an angle upon impact that fell outside the acceptable range for facilitating its attachment.

One notable result that emerged was the success rate of object #2, which stood out among the others. Despite having physical properties comparable to object #4 and object #5, object #2 exhibited a lower rate than both, which is 77.5 %. A potential explanation for this difference can be attributed to the presence of two additional steel balls in object #2, which alters the required range of contact angles due to increased mass, which causes it to rotate more during flight resulting in a steeper angle of contact upon impact. This observation aligns with our data, as object #4, which is also heavier, demonstrates a lower success rate compared to object #1. Interestingly, object #5 shows a 100% success rate, contrary to our expectations based on its similarities to object #2. We hypothesize that the thicker disc in object #5 led to an increased radius of mid-air rotation, allowing the dart to rotate less within the same distance. Consequently, this reduced the angle of contact, resulting in a higher success rate.

Furthermore, we observed that shifting the center of mass due to additional weight at the dart head posed challenges in learning the dart's mid-air dynamics. Additionally, the presence of real system noise had an impact on the overall success rate of the throwing process.

E. Limitations

This work exhibits certain limitations that have been addressing to a certain extent. Firstly, while the current approach achieves high performance, it lacks the ability to learn the mid-air dynamics of the object. This limitation suggests that the method may not fully capture the complex dynamics and aerodynamic interactions occurring during the dart's flight. Secondly, the target distance in current work is fixed. This fixed distance may limit the generalizability of the approach to different throwing scenarios where variable target distances are encountered. Furthermore, the applicability of the current method is limited to non-cluttered environments, as it does not incorporate strategies for grasping and throwing objects in cluttered scenarios.

V. CONCLUSION

In this work, we have developed and implemented the TT-RL framework to learn overhand throwing skills on a real

robot, specifically focusing on accurate throwing of small objects comprising a deformable shaft and a rigid head. The main challenge we address is acquiring precise throwing ability for these small objects, which undergo unpredictable spins in the air after being released by the robot. Even small variations in the physical properties of the objects, such as mass, center-of-mass, and length, can significantly affect their trajectory and their final pose as they approached the target location. Through the utilization of high-resolution tactile sensing, we have successfully trained the robot to achieve impressive throwing performance. Our method demonstrated the capability for generalization, as we tested the learned policy on a diverse set of objects. This robustness suggests the potential of our approach for real-world applications.

Moving forward, future work could involve expanding the learning framework to include different target locations, enabling the robot to throw accurately at specific areas. Additionally, leveraging tactile feedback during in-hand manipulation and object release poses could provide deeper insights into the object's dynamics in the air, leading to further refinement of the throwing technique.

REFERENCES

- [1] K. John E., "Throwing, the shoulder, and human evolution," *American journal of orthopedics (Belle Mead, N.J.)*, vol. 45, pp. 110–114, 2016.
- [2] A. Zeng, S. Song, J. Lee, A. Rodriguez, and T. Funkhouser, "Tossingbot: Learning to throw arbitrary objects with residual physics," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1307–1319, 2020.
- [3] F. Raptopoulos, M. Koskinopoulou, and M. Maniadakis, "Robotic pick-and-toss facilitates urban waste sorting," in *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pp. 1149–1154, 2020.
- [4] J. Kober, K. Muelling, and J. Peters, "Learning throwing and catching skills," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5167–5168, 2012.
- [5] M. Monastirsky, O. Azulay, and A. Sintov, "Learning to throw with a handful of samples using decision transformers," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 576–583, 2023.
- [6] C. Wang, S. Wang, B. Romero, F. Veiga, and E. Adelson, "Swingbot: Learning physical features from in-hand tactile exploration for dynamic swing-up manipulation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5633–5640, 2020.
- [7] Z. Xu, J. Wu, A. Zeng, J. B. Tenenbaum, and S. Song, "Densephysnet: Learning dense physical object representations via multi-step dynamic interactions," 2019.
- [8] M. Mason and K. Lynch, "Dynamic manipulation," in *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '93)*, vol. 1, pp. 152–159 vol.1, 1993.
- [9] Y. Gai, Y. Kobayashi, Y. Hoshino, and T. Emaru, "Motion control of a ball throwing robot with a flexible robotic arm," *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, vol. 7, pp. 937–945, 2013.
- [10] J.-S. Hu, M.-C. Chien, Y.-J. Chang, S.-H. Su, and C.-Y. Kai, "A ball-throwing robot with visual feedback," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2511–2512, 2010.
- [11] W. Mori, J. Ueda, and T. Ogasawara, "1-dof dynamic pitching robot that independently controls velocity, angular velocity, and direction of a ball: Contact models and motion planning," in *2009 IEEE International Conference on Robotics and Automation*, pp. 1655–1661, 2009.
- [12] J. U. Wataru Mori and T. Ogasawara, "A 1-d.o.f. dynamic pitching robot that independently controls velocity, angular velocity and direction of a ball," *Advanced Robotics*, vol. 24, no. 5-6, pp. 921–942, 2010.
- [13] D. M. Lofaro, R. Ellenberg, P. Oh, and J.-H. Oh, "Humanoid throwing: Design of collision-free trajectories with sparse reachable maps," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1519–1524, 2012.
- [14] H. Miyashita, T. Yamawaki, and M. Yashima, "Parts assembly and sorting by throwing manipulation: Planning and control," *Journal of System Design and Dynamics*, vol. 5, no. 1, pp. 139–154, 2011.
- [15] T. Senoo, A. Namiki, and M. Ishikawa, "High-speed throwing motion based on kinetic chain approach," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3206–3211, 2008.
- [16] H. Miyashita, T. Yamawaki, and M. Yashima, "Control for throwing manipulation by one joint robot," in *2009 IEEE International Conference on Robotics and Automation*, pp. 1273–1278, 2009.
- [17] T. Orion and R. Alberto, "Optimal shape and motion planning for dynamic planar manipulation," *Autonomous Robots*, vol. 43, no. 2, pp. 327–344, 2019.
- [18] A. Takahashi, M. Sato, and A. Namiki, "Dynamic compensation in throwing motion with high-speed robot hand-arm," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6287–6292, 2021.
- [19] J. H. Kim, "Optimization of throwing motion planning for whole-body humanoid mechanism: Sidearm and maximum distance," *Mechanism and Machine Theory*, vol. 46, no. 4, pp. 438–453, 2011.
- [20] N. Govindan, B. Ramachandran, P. H. V. Sai, and K. M. Krishna, "A novel hybrid gripper capable of grasping and throwing manipulation," *IEEE/ASME Transactions on Mechatronics*, vol. 28, no. 6, pp. 3317–3328, 2023.
- [21] A. Sintov and A. Shapiro, "A stochastic dynamic motion planning algorithm for object-throwing," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2475–2480, 2015.
- [22] Y. Liu, A. Nayak, and A. Billard, "A solution to adaptive mobile manipulator throwing," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1625–1632, 2022.
- [23] C. Chi, B. Burchfiel, E. Cousineau, S. Feng, and S. Song, "Iterative residual policy: For goal-conditioned dynamic manipulation of deformable objects," *The International Journal of Robotics Research*, vol. 0, no. 0, p. 02783649231201201, 0.
- [24] D. Bianchi, G. Antonelli, C. Laschi, and E. Falotico, "Open-loop control of a soft arm in throwing tasks," *19th International Conference on Informatics in Control, Automation and Robotics*, pp. 138–145, 01 2022.
- [25] A. Ghadirzadeh, A. Maki, D. Krägic, and M. Björkman, "Deep predictive policy training using reinforcement learning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2351–2358, 2017.
- [26] Y. Ji, Z. Li, Y. Sun, X. B. Peng, S. Levine, G. Berseth, and K. Sreenath, "Hierarchical reinforcement learning for precise soccer shooting skills using a quadrupedal robot," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1479–1486, 2022.
- [27] H. Kasaei and M. Kasaei, "Throwing objects into a moving basket while avoiding obstacles," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3051–3057, 2023.
- [28] D. Bianchi, M. G. Antonelli, C. Laschi, A. M. Sabatini, and E. Falotico, "Softtoss: Learning to throw objects with a soft robot," *IEEE Robotics & Automation Magazine*, pp. 2–12, 2023.
- [29] Y. She, S. Wang, S. Dong, N. Sunil, A. Rodriguez, and E. Adelson, "Cable manipulation with a tactile-reactive gripper," *The International Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1385–1401, 2021.
- [30] S. Wang, Y. She, B. Romero, and E. Adelson, "Gelsight wedge: Measuring high-resolution 3d contact geometry with a compact robot finger," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6468–6475, 2021.
- [31] I. H. Taylor, S. Dong, and A. Rodriguez, "Gelslim 3.0: High-resolution measurement of shape, force and slip in a compact tactile-sensing finger," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 10781–10787, IEEE, 2022.
- [32] Y. Du, G. Zhang, Y. Zhang, and M. Y. Wang, "High-resolution 3-dimensional contact deformation tracking for fingervision sensor with dense random color pattern," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2147–2154, 2021.
- [33] G. Zhang, Y. Du, H. Yu, and M. Y. Wang, "Deltact: A vision-based tactile sensor using a dense color pattern," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10778–10785, 2022.
- [34] F. R. Hogan, M. Bauza, O. Canal, E. Donlon, and A. Rodriguez, "Tactile regrasp: Grasp adjustments via simulated tactile transformations," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2963–2970, 2018.
- [35] Y. Chebotar, K. Hausman, Z. Su, G. S. Sukhatme, and S. Schaal, "Self-supervised regrasping using spatio-temporal tactile features and

- reinforcement learning,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1960–1966, 2016.
- [36] R. Calandra, A. Owens, D. Jayaraman, J. Lin, W. Yuan, J. Malik, E. H. Adelson, and S. Levine, “More than a feeling: Learning to grasp and regrasp using vision and touch,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3300–3307, 2018.
- [37] S. Fujimoto, H. van Hoof, and D. Meger, “Addressing function approximation error in actor-critic methods,” *CoRR*, vol. abs/1802.09477, 2018.
- [38] W. Yuan, S. Dong, and E. H. Adelson, “Gelsight: High-resolution robot tactile sensors for estimating geometry and force,” *Sensors*, vol. 17, no. 12, 2017.
- [39] GelSight, “GelSight Mini.” <https://www.gelsight.com/gelsightmini/>.
- [40] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dормann, “Stable-baselines3: Reliable reinforcement learning implementations,” *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021.
- [41] OpenAI, “TD3.” <https://spinningup.openai.com/en/latest/algorithms/td3.html>.



Hongyu Yu received the B.S. and M.S. degrees in Electronics Engineering from Tsinghua University, China, in 1997 and 2000, respectively, and the Ph.D. degree in Electrical Engineering from the University of Southern California, USA, in 2005. He was a postdoctoral research associate at the University of Southern California from 2005–2007. He joined Arizona State University (ASU), USA, in 2008 holding as assistant professor (2008–2014) and associate professor (2014–2017) in both School of Earth and Space Exploration and School of Electrical, Computer and Energy Engineering. During the tenure at ASU, he was the principal investigator for several projects from NASA, NSF and Intel. He has joined the Hong Kong University of Science and Technology, Hong Kong since new year day of 2018. He is currently serving on the Faculty of Department of Mechanical and Aerospace Engineering as an associate professor. His interest is providing engineering solutions for scientific studies and real-life demands. His research focuses on robotics, automation, smart structures, sensors and sensing platforms, and miniature instrumentation with applications for cube satellites, aerial vehicles and consumer electronics.



Shoaib Aslam is a Ph.D. student in the Department of Mechanical and Aerospace Engineering at the Hong Kong University of Science and Technology, Hong Kong. He also joined Mechanisms and Robotic Systems Lab at Purdue University, USA, as a Ph.D. research exchange student in 2022–23. Prior to that, he received the B.Sc. in Mechatronics Engineering and M.Sc. in Mechanical Engineering from University of Engineering and Technology, Taxila, Pakistan, in 2012 and 2016, respectively. He has joined the Department of Mechatronics and Control Engineering at University of Engineering and Technology, Lahore, Pakistan, as a lecturer in 2014 and is currently on study leave. His research interest includes robotic grasping and manipulation, sensory perception, and machine learning.



Michael Yu Wang (Fellow, IEEE) received the B.E. degree in Mechanical and Manufacturing Engineering from Xi'an Jiaotong University, Xi'an, China, in 1982, the M.S. degree in Engineering Mechanics from Pennsylvania State University, State College, USA, in 1985, and the Ph.D. degree in Mechanical Engineering from Carnegie Mellon University, Pittsburgh, USA, in 1989. He is currently a Chair Professor and Dean of School of Engineering, Great Bay University, Dongguan, China. Before joining Great Bay University, he served on the Faculty of the Department of Mechanical and Aerospace Engineering, Monash University, Melbourne, VIC, Australia, as Head of the Department, the Hong Kong University of Science and Technology as the Director of the Robotics Institute, the Mechanical Engineering Faculty of the University of Maryland, the Chinese University of Hong Kong, and the National University of Singapore. His research interests include robotics, automation, multifunctional structures, and additive manufacturing, with more than 300 technical publications in these areas. Prof. Wang is a fellow of ASME and HKIE. He is the Editor-in-Chief Emeritus of IEEE Transactions on Automation Science and Engineering and an Editor for Computer-Aided Design, Structural and Multidisciplinary Optimization, and Frontiers of Mechanical Engineering.



Krish Kumar is an undergraduate student, pursuing a Bachelor of Science in Computer Engineering with a specialization in Machine Learning and Artificial Intelligence at Purdue University, United States. His research interest includes robotic manipulation, human-robot interactions, and machine learning.



Pokuang Zhou is a Ph.D. student at Purdue University, United States. Prior to that, he received his bachelor's and master's degree in Engineering from Huazhong University of Science and Technology. His research, at the intersection of control theory, machine learning, human-robot interaction, multimodal robotic systems and robotic manipulation.



Yu She is an assistant professor at Purdue University School of Industrial Engineering. Prior to that, he was a postdoctoral researcher in the Computer Science and Artificial Intelligence Laboratory at MIT from 2018 to 2021. He earned his Ph.D. degree in the Department of Mechanical Engineering at the Ohio State University in 2018. His research, at the intersection of mechanical design, sensory perception, and dynamic control, explores human-safe collaborative robots, soft robotics, and robotic manipulation.